

```
#####
#
FractionToLocation.1 <- function(system, numerator, denominator){
#
#   system = 1   Calkin-Wilf
#   system = 2   driB
#   system = 3   Yu-Ting
#   system = 4   Yurramendi-1
#
# Given a fraction numerator / denominator,
# by tracking until the the root (numerator = 1, denominator = 1)
#
# Tracking is related to the Euclid's algorithm [18]
#
track <- vector()
#
while(numerator / denominator != 1){
  if(numerator / denominator < 1){
    if(system == 1 | system == 3){
      denominator <- denominator - numerator
    }
    if(system == 2 | system == 4){
      temp <- numerator
      numerator <- denominator - temp
      denominator <- temp
    }
    track <- c(track, 0)
  }
  if(numerator / denominator > 1){
    if(system == 1 | system == 4){
      numerator <- numerator - denominator
    }
    if(system == 2 | system == 3){
      temp <- numerator
      numerator <- denominator
      denominator <- temp - denominator
    }
    track <- c(track, 1)
  }
}
track <- c(track, 1)
#
# Tracking back from the root (1,1)
# to the given fraction
#
location <- sum(track*2^(0:(length(track)-1)))
#
return(location)
#
}
#
#####
#
# Examples: Compute location of 9/5 in all four systems
#
FractionToLocation.1(1, 9, 5) ### 61
FractionToLocation.1(2, 9, 5) ### 55
FractionToLocation.1(3, 9, 5) ### 39
FractionToLocation.1(4, 9, 5) ### 57
#
#####
```